

Discovery of Jumping Emerging Patterns Using Genetic Algorithm

Sumera Qurat ul Ain ^{#1}, Saif ur Rehman ^{*2}

UIIT, Arid Agriculture University,

Rawalpindi, Pakistan

¹sumera_quratulain@yahoo.com

UIIT, Arid Agriculture University,

Rawalpindi, Pakistan

²saif@uaar.edu.pk

Abstract— Patterns that only occur in objects belonging to a single class are called **Jumping Emerging Patterns (JEP)**. JEP based Classifiers are considered one of the successful classification systems. Due to its comprehensibility, simplicity and strong differentiating abilities JEPs have captured significant recognition. However, discovery of JEPs in a large pattern space is normally a time consuming and challenging task because of their exponential behaviour. In this work a novel method based on genetic algorithm (GA) is proposed to discover JEPs in large pattern space. Since the complexity of GA is lower than other algorithms, so we have combined the power of JEPs and GA to find high quality JEPs from datasets to improve performance of classification system. Our proposed method explores a set of high quality JEPs from pattern search space unlike other methods in literature that compute complete set of JEPs, Large numbers of duplicate and redundant JEPs are filtered out during their discovery process. Experimental results show that our proposed Genetic-JEPs are effective and accurate for classification of a variety of data sets and in general achieve higher accuracy than other standard classifiers.

Keywords- Classification, Jumping Emerging Patterns, Genetic Algorithm.

I. INTRODUCTION

One of the primary objectives in data mining is classification and has been researched broadly in the fields of expert systems [1], machine learning [6], and neural networks [7] over decades. With reference to machine learning classification is the procedure to explore classifiers or predictive models that can differentiate testing data between different classes, in model derivation first training data is examined on the basis of quantifiable feature set (types include ordinal, categorical, integer-valued and real-valued) [8]. This training data contains information about the class of each instance from which it belongs. An essential part of the classification is termed as “classifiers” which is an implementation of an algorithm to classify unlabeled instances or a mathematical function which assigns test instances, to a category or class.

There exist problems when high accuracy is required along with explanation on which basis; it classifies objects to a class so that it should be understandable by users [2]. Many

classifiers lack such explanations, which is an important drawback to use them. For Example, in US Equal credit opportunity act, if a credit has been denied by a financial institution, an explanation also required explaining the reasons on which ground credit has been rejected to an applicant, if such reasons are not explained properly, credit denied considered illegal [4]. Medical diagnosis and mineral prospect ions are some other fields to name a few, where explanations and clarification are key user requirements in the classifiers [3]. A family in understandable classifiers are develop from emerging patterns [5]. “A pattern can be defined as an expression” in a language to describe a group of objects[7], moreover emerging pattern (EP) occurs abundantly in objects in one class except rare to locate in objects of other classes [9]. Jumping Emerging Pattern (JEP) is a type of EPs, are Itemset whose support rises suddenly from zero in one class of data to nonzero in another class, the ratio of support rises to infinite [1]. Due to their firm predictive power JEPs are extensively used in emerging pattern based classifiers [10]. Moreover JEPs are combination of simple conditions (e.g [Color = red] AND [Gender = female] AND [Age > 26]) therefore they are easily understandable by the user [11].

With all their merits of JEPs based classification, major drawback lies in the number of discovered patterns, useful for classification. If data has noise, number of patterns becomes larger. Searching useful JEPs in training data is a key procedure in JEP based classification system. Previous techniques either uses support threshold value or predefined numbers of patterns [12]. Due to these problems JEPs discovery is considered as a challenging task. To avoid redundancy in JEPs, we uses a subset of JEPs called minimal JEPs. Genetic algorithm adopts global search method that is better in finding solution, particularly in large search spaces. Since, time complexity of GA is less than other algorithms [13], so we have combined the power of JEPs and GA to find significant JEPs from different data sets for classification of data.

In our work we propose a new method to discover minimal JEPs that have significant affect on classification accuracy. Exploring fittest JEPs from datasets is a multistep process. In

first step, chromosomes(candidate solutions) are initiated these chromosomes are termed as population in literature, in subsequent steps solution of each population are used to generate new population in next stage having hope to get better population in each generation when compare with previous ones. This population generation is regulated with the help of a fitness function whole process is repeated until terminating conditions reached. After complete iterations of genetic algorithm, we will get a high quality fittest JEPs list, which will be evaluated on some test data for classification accuracy.

This paper is organized as: section II presents some background knowledge and related work. Section III presents proposed algorithm for discovery of JEPs. Section IV presents experimental results and analysis and section V describes conclusion.

II. BACKGROUND AND RELATED WORK

This section consists of some basic concepts and definitions, used throughout in this paper. A dataset is a set of objects having multiple attributes (A_1, A_2, \dots, A_n), each data object is referred as an instance. Each instance of dataset has associated class $C \in \{C_1, C_2, \dots, C_n\}$ [14]. If I represents set of items or patterns, P is an itemset which is subset of I , if any instance S holds an itemset P provided $P \subseteq S$ then support of an itemset P in a data set D represent as $supp_D(P)$ is $count_D(P)/|D|$, Where $Count_D(P)$ is the number of instances in D containing P , and $|D|$ denoted total instances in D . Growth Rate is the ratio of support of a pattern in its native class C_p with the support of pattern in other class, it represents the predictive power of the pattern. $Growth Rate(P)$ is defined as:

$$Growth Rate(P) = \begin{cases} 0, & \text{if } support(P, C) = 0 \wedge support(P, C_p) = 0 \\ \infty, & \text{if } support(P, C) = 0 \wedge support(P, C_p) > 0 \\ \max\left(\frac{support(P, C_p)}{support(P, C)}, \frac{support(P, C)}{support(P, C_p)}\right), & \text{otherwise} \end{cases}$$

Definition 1. Let D is a dataset comprises on D_1 and D_2 where objects of D_1 belongs to one class and objects of D_2 belongs to a different class, a Jumping Emerging Pattern (JEP) from D_1 to D_2 is an itemset X , that satisfies the condition:

$$supp_{D_1}(X) = 0 \text{ and } supp_{D_2}(X) > 0$$

Definition 2. If a JEP does not hold another JEP as a proper subset, then it is called minimal JEP.

Example 1. Let For dataset presented in Table I, itemsets $\{l, p\}(1 : 0)$, $\{l, n, p\}(1 : 0)$, $\{l, o, p\}(1 : 0)$, $\{l, n, o, p\}(1 : 0)$, $\{m, p\}(2 : 0)$, $\{m, n, p\}(1 : 0)$, $\{m, o, p\}(1 : 0)$, $\{m, n, o, p\}(1 : 0)$, and $\{n, o, p\}(2 : 0)$ are JEPs of class 1; itemsets $\{l, m\}(0 : 2)$, $\{l, m, n\}(0 : 1)$, $\{l, m, o\}(0 : 1)$, and $\{l, m, n, o\}(0 : 1)$ are JEPs of class 2. The total number of JEPs is 13, and four among them are minimal JEPs, namely, $\{l, p\}(1 : 0)$, $\{m, p\}(2 : 0)$, $\{n, o, p\}(2 : 0)$, and $\{l, m\}(0 : 2)$ [1].

In available literature, researchers proposed several JEP-based Classifiers [15]. In [16] concept of Essential Jumping Emerging pattern (eJEPs) is proposed, this classifier utilizes

less JEPs than the other JEP classifier. Limitations of this research includes that it is not complete not sound to discover all JEPs and yields itemsets other than actual JEPs [2]. Classifier proposed in [2] is a FP tree based technique which

TABLE I
Example dataset having two classes

ID	Class Labels	Instances (Itemsets)
1	D_1	$\{l, n, o, p\}$
2	D_1	$\{l\}$
3	D_1	$\{m, p\}$
4	D_1	$\{m, n, o, p\}$
5	D_2	$\{l, m\}$
6	D_2	$\{n, p\}$
7	D_2	$\{l, m, n, o\}$
8	D_2	$\{o, p\}$

uses a FP growth algorithm to explore JEPs, exploring tree based structure has high time and space complexity when comparing with other data structures, in [1] Strong JEP is proposed, authors claim that SJEPs achieve higher quality when comparing with JEPs, it is an efficient classifier due to the less number of SJEPs. In same research author proposed Noise tolerant emerging patterns (NEPs) and generalized noise tolerant emerging patterns (GNEPs). The limitations of these JEPs include that they find SJEPs in two steps initially it generate a large number of JEPs then in second step generated JEPs are filtered out to produce a small set of SJEPs. This twostep process is time consuming and its time and space complexity is higher than its competitors in literature, another major demerit of this study is its requirement of minimum threshold value to prune large set of JEPs. In [17] negJEP-Classifier and JEPN-classifier introduced, these classifiers use negative information for classification, results revealed that a limitation of this research is its infeasibility to build JEPN classifier for some data sets [17]. [18] discover Top K minimal jumping emerging patterns, proposed method finds strong JEPs which are CP tree based, instead of exploring all JEPs it reduces search space and explore top K - JEPs only on the bases of increasing minimum threshold support value. The drawback of this research includes additional pattern counting [18]. In [19] Highest impact JEPs introduced, which is based on introducing a new coefficient REAL/ALL, this coefficient is helpful in comparing discriminative power of distinguished JEPs collections, demerit of this research is its limited application to specific datasets. An improved tree based method is proposed in [14], this method explores SJEPs more efficiently, drawback of this method include it cannot mine SJEPs directly, [20] extended border based mining method and proposed a classifying model for discovering JEPs with occurrence count for classification. The drawback of this method which can be improved is its discovery of a large number of JEPs. The scheme presented in [21] is based on CP tree whose growth rate is dynamically increasing, it prunes the search space with the help of new pattern pruning method, demerits of this research is like in every tree based search is its space complexity and inability to handle more complex patterns [21]. In [12] a method for discovery of minimal JEPs introduced and to the best of my knowledge I found the last

reported work on JEPs is the use of cosine similarity with JEPs in classification by [10] but its accuracy is compromised.

Various methods have been proposed in the past to reduce the huge pattern set but it is still an open research area to considerably reduce the number of derived patterns and improve the worth of selected patterns.

III. PROPOSED ALGORITHM FOR DISCOVERY OF JEPs

Inspired by Darwin's Evolution theory Genetic algorithm is a solution to many problems. Genetic algorithm adopts global search strategy that is better for searching optimized solution, particularly in bigger search spaces. In the following section, detailed procedure of discovery of JEPs by using Genetic Algorithm is given, major steps includes: Individual representation, Initial Population Generation, Fitness evaluation, creation of next generation by using selection, reproduction, crossover and mutation.

A. Individual Representation

Different type of attributes exists, e.g. categorical, numerical, in numerical attributes an assumption is made, this assumption is about the discretized range of value in intervals. Different combinations of attribute and their values are normally used to express patterns, like (*Color = red, gender = female, Age = 26*) or as logical properties, like [*Color = red*] AND [*Gender = female*] AND [*Age > 26*]. Normally, data need to be encoded into chromosomes because genetic algorithm cannot directly handle data in the solution search space. Every Chromosome represents a candidate pattern and have the form of ($A_1 = V_{1j} \wedge \dots \wedge A_n = V_{nj}$). Where A_i represents i th feature and V_{ij} represents j th value of the i th feature's domain and n represents length of chromosome [22].

A_1	A_2	A_3	A_4	A_5	A_n
V_{1j}	V_{2j}	V_{3j}	V_{4j}	V_{5j}	V_{nj}

Fig. 1. Individual Representation

B. Initial Population Generation

We have randomly created population of n chromosomes (where n is the number of chromosomes in population), population size can be fixed according the problem. In our method we have used population of 100 chromosomes. These chromosomes represent patterns by randomly selecting attributes and their values.

C. Fitness Evaluation of Individual

When applying Genetic Algorithm fitness function is the central part that plays a vital role in problem optimization. We need to find some measures (*support* and *growth ratio* described in section II) of these patterns to analyze their differentiating power between different classes. After calculating the *support* and *growth ratio* values of a pattern proposed algorithm will check the pattern whether it is a JEP or not? In our method we are exploring Minimal JEPs so extra attributes will be removed by applying a pruning method. Growth ratio of a pattern will be its Fitness value. Having very low fitness value patterns will be less probable to survive for

the next generation. Class will be assigned to the pattern according to its growth rate. Then discovered Minimal JEPs will be added to a global pattern list. Each time when we add a discovered pattern to the pattern list there is a check to avoid duplication of pattern in global pattern list.

D. Create Next generation

After fitness evaluation we will create new population for the next generation by repeating following steps unless new population according to population size is completed.

1) *Selection*: In selection process those patterns are used which has good fitness values to produce next generation with the expectation to have offspring that has higher fitness value than their parents. In order to getting better members we have used Proportional Roulette wheel selection. Each chromosome has selection probability i.e. directly proportional with individual's fitness value. Chromosomes with higher probability capture the larger fragment; while the less fit captures likewise smaller fragment in the roulette wheel. Clearly those chromosomes which have larger fragment size have more chances to be chosen as parent for next generation. Let f_1, f_2, \dots, f_n be fitness values of individual 1, 2, ..., n . Then the selection probability, P_i for individual i can be calculated as[23]:

$$P_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

2) *Crossover*: Procedure of crossover begins with selecting more than two parent solutions (patterns) and generating two child solutions from them. We have used single point crossover so first we select a random point then beyond that point all data in both patterns is swapped between the two parents. This results in two offspring patterns.

3) *Mutation*: Mutation will be applied on a single chromosome at a time. It preserves gene diversity in population and ensures searching in the entire solution space. In our proposed method mutation rate is very low i.e 5 % because we don't want major changes in population it is used only for global optimization. Complete algorithm is given as follows:

Algorithm 1: Algorithm for discovery of JEPs using GA

Input: D : $m \times n$ matrix, training dataset.

GenNo : maximum number of allowed generations

MR : Mutation rate

CR : Crossover Rate

PSize : Maximum number of chromosomes in

Population

Output: The best Jumping emerging pattern in all generations.

1. Generate random population of n chromosomes , where $n = 1 \dots PSize$.
2. **for** $i = 1$ to GenNo **do**
3. **for** $j = 1$ to PSize **do**
4. Calculate fitness of each individual.
5. Add discovered patterns to list.
6. Prune pattern list.

7. Calculate selection probability of each individual.
8. **end do**
9. Create fresh population through repeating the following steps unless the population completion:
 10. Selection();
 11. Crossover();
 12. Mutation();
 13. Add new child to fresh population.
 14. Use newly created population for further running of algorithm.
15. **Check Termination criteria:** If the terminating criterion is satisfied, **stop**, Termination condition is predefined number of generations. If termination criterion is not satisfied then go to **step 2**.
16. **end for**
17. Evaluate Pattern list on test set.
18. Display Result.
19. **end**

Classification may be binary or multiclass. In binary classification, test objects can be classified in two classes only, whereas in multi-class more than two classes can be assigned to test objects. Our proposed method can be used for multiclass classification. For multiclass classification we will use one against all technique.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

Databases used in the experiments are downloaded from the UCI ML repository. Description of Datasets is given in the table II. Results of all the experimental are expressed as Average, i.e. ratio of correct classified instances with the overall number of instances present in database. To ensure accuracy CV-10 methodology is used. Results represent the average classification performance over the 10 folds. Moreover, frequency of experiment execution is five. To ensure robustness in the values, results of all five executions are than used to calculate average.

TABLE II
Description of Datasets

Datasets	# of Instances	# of attributes	# of classes
adult+stretch data	20	5	2
breast-cancer	286	10	2
Crx	690	16	2
Diabete	768	9	2
Glass	214	10	6
Hayes Roth	132	5	3
Iris	150	5	3
Monks	432	7	2
Nursery	12960	9	5
Sonar	208	61	2
Soybean-small	47	22	4
Tic Tac Toe	958	10	2
Vehicle	846	19	4
Xad	94	19	4
Xae	94	19	4

It is evident that all these experiments are aimed to assess the success and predictive power of the discovered JEPs and its discovery method. For the experiments of the classifier, 15 common and benchmark databases for classification are used. Proposed technique is implemented in Microsoft Visual C# .Net Framework 4.0, using Microsoft Visual Studio 2015. All experiments were conducted on HP ProBook 4530 Core i5-2430M CPU @ 2.40 GHz, 4Gb RAM) running Windows 7 Home Premium 64 bit version. Statistical details on important information of the data set used for the experiments are given in the following table.

In experiments, different parameters have been used for the evaluation purposes. These parameters are included *GenNo* represents the Total Number of Generation used for the iteration of the proposed algorithm. Other parameters are included *MR* which is the mutation rate for global optimization. *CR* is the Cross Over Rate which is used for the generation of the next population. *PSize* is the size of population. Finally, the *D* parameter is used for training dataset given to proposed algorithm. We have used different values for the above mentioned parameters. These parameter values are given in the Table III.

TABLE III
Experimental Parameters

Parameter	Value
GenNo	50
MR	0.15
CR	0.85
PSize	100
D	M X N matrix training data set

Genetic-JEPs (G-JEPs) accuracy has been compared with the seven popular classifiers, i.e. Naïve Bayes, J48, ID3, Zero-R, One- R, bagging, and Random Forest. The accuracy of Naïve Bayes, J48, ID3, Zero-R, One- R, bagging, and Random Forest are obtained using Weka 3.6.9 implementation.

TABLE IV
G-JEPs Accuracy on various datasets

Datasets	G-JEPs Accuracy
adult+stretch data	100.00
breast-cancer	96.494
Crx	85.285
Diabete	88.34
Glass	59.345
Hayes-roth	100.00
Iris	95.066
Monks	99.459
Nursery	90.848
Sonar	57.211
Soybean-small	100.00
tic-tac-toe	98.121
vehicle	56.146
xad	71.276
xae	67.021

We present a best/ same / poor summary in table VI to evaluate performance of the G- JEPs against competitors. best/same/poor represents the number of data sets on which

the G- JEPs results greater accuracy than others, the number for which the both classifier results same accuracy, and the number on which comparator achieves greater accuracy. From table V and table VI, it is clearly shown that the G-JEPs obtained significantly better accuracy on majority datasets.

The G-JEPs classifier has the best performance in 10 out of 15 datasets. Moreover our proposed method has not proof as a worst performer among its competitors.

TABLE V
Accuracy comparison with standard classifiers

Datasets	Naive Bayes	J48	bagging	ID3	Random Forest	ZeroR	One-R	G-JEPs
adult+stretch_data	100.00	100.00	100.00	100.00	100.00	60.00	70.00	100.00
breast-cancer	72.08	75.524	67.832	56.993	69.231	70.279	65.734	96.494
crx	84.787	85.36	85.942	72.029	83.913	55.507	85.507	85.285
diabete	72.91	74.218	73.046	63.411	69.531	65.104	74.088	88.34
glass	63.081	59.813	63.084	61.215	69.53	35.514	44.392	59.345
Hayes-roth	80.303	72.727	77.272	62.121	77.272	37.878	43.930	100.00
iris	87.333	90.667	90.00	89.333	90.00	33.333	82.667	95.066
monks	75.00	96.527	98.6111	95.370	96.527	49.074	75.00	99.459
nursery	90.324	97.052	97.268	98.186	97.938	33.333	70.972	90.848
sonar	67.788	71.153	77.884	---	79.326	53.365	62.50	57.211
Soybean-small	100.00	97.872	100.00	95.744	100.00	36.170	87.234	100.00
tic-tac-toe	69.624	85.073	90.710	83.298	90.709	65.344	69.937	98.121
vehicle	57.10	65.484	67.021	61.820	65.484	25.650	41.489	56.146
xad	56.383	53.191	48.936	38.297	57.446	30.851	41.489	71.276
xae	57.447	53.191	48.936	38.297	57.446	30.851	41.489	67.021

TABLE VI
Best/ same/ poor record of G-JEPs vs. Alternatives

JEPs Vs	Naive Bayes	J48	bagging	ID3	Random Forest	0- R	1- R
best/same/poor	10,2,3	9,1,5	8,2,5	11,1,3	9,2,4	15,0,0	13,0,2

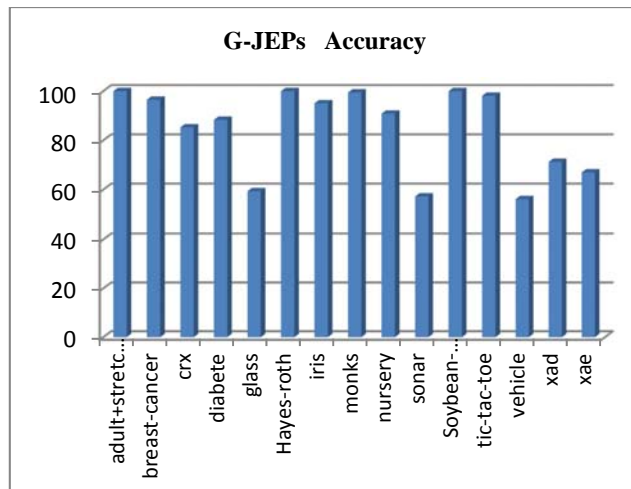


Fig. 2. G-JEPs accuracy on various UCI datasets

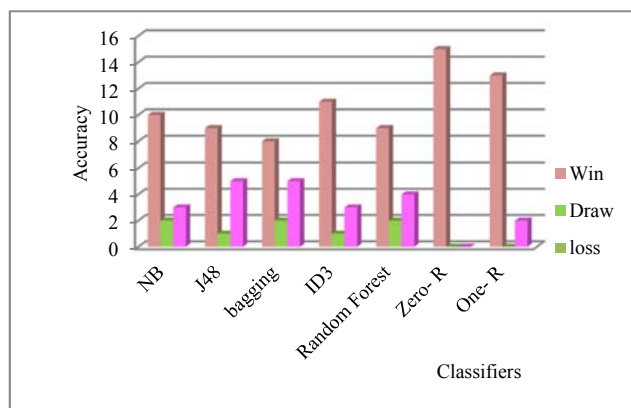


Fig. 3. Best/ Same/ Poor performance of G-JEPs vs Alternatives

V. CONCLUSION

A comprehensive study about Jumping Emerging Patterns (JEPs) and the earlier related research works has been presented to highlight their limitations in order to address classification problems. Our contribution presented in this work is a novel method for JEPs discovery using Genetic Algorithm aimed to overcome the deficiencies of the previous related approaches in this domain. To decrease the number of JEPs, we used a function for pruning. Large numbers of duplicate and redundant JEPs are filtered out during their discovery process. We developed an accurate and effective classification model based on JEPs. We have performed experiments on standard data sets to illustrate that proposed G-JEPs are effective and accurate for classification of a variety of data sets and in general achieves higher accuracy than other standard classifiers.

REFERENCES

- [1] H. Fan, and k. Ramamohanarao. 2006. Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers, *knowledge and Data Engineering, IEEE Transactions on*, vol. 18, pp.721-737 2006.
- [2] J. Bailey, T. Manoukian, and K. Ramamohanarao, "Fast Algorithms for Mining Emerging Patterns", *Proc. Sixth European Conf. Principles and Practice of knowledge Discovery in Databases(PKDD'02)*, 2002.
- [3] M. Garcia-Borroto, J. F. Martinez-Trinidad and J. A. Carrasco-Ochoa. 2014. A survey of emerging patterns for supervised classification, *Artificial Intelligence Review*, 42(4), pp. 705-721.
- [4] M. Garcia-Borroto, J. F. Martinez-Trinidad, J. A. Carrasco-Ochoa, M.A. Medina-Pérez, and J. Ruiz-Shulcloper. 2010. LCMine: An efficient algorithm for mining discriminative regularities and its application in supervised classification. *Pattern Recognition*, 43(9), pp. 3025-3034.
- [5] K. Ramamohanarao and H. Fan. 2007. Patterns based classifiers, *World Wide Web*, 10(1), pp.71-83.
- [6] T.M. Mitchell, *Machine Learning*. McGraw-Hill Higher Education, 1997.
- [7] G. Pateski, and W.Frawley. 1991. *Knowledge discovery in databases*, MIT press.
- [8] Y. Ma, L. Bing and H.Wynne. 1998. Integrating classification and association rule mining, *In Proceeding of the fourth international conference on knowledge discovery and data mining*.
- [9] P. Andruszkiewicz. 2011. Lazy approach to preserving classification with emerging patterns, *In Emerging intelligent technologies in industry*, pp. 253-268. Springer Berlin Heidelberg.
- [10] M. Ferrandin, A. Boava, and A.S. R. Pinto. 2015. Classification Using Jumping Emerging Patterns and Cosine Similarity. *In Proceedings on the International Conference on Artificial Intelligence (ICAI)* pp. 682.
- [11] G. Dong and J. Li. Efficient mining of emerging patterns: discovering trends and differences. In: *KDD*, pp.43-52n(1999).
- [12] B. Kane, B. Cuissart and B. Cremilleux. 2015. Minimal Jumping Emerging Patterns: Computation and Practical Assessment, *In Advances in Knowledge Discovery and Data Mining*, pp. 722-733. Springer International Publishing.
- [13] M. Kabir, M. J., S. Xu, B. H. Kang, and Z. Zhao. 2015. Comparative analysis of genetic based approach and Apriori algorithm for mining maximal frequent item sets. *In IEEE Congress on Evolutionary Computation (CEC)*, pp. 39-45.
- [14] Chen, Xiangtao, and Lijuan Lu. "An improved algorithm of mining Strong Jumping Emerging Patterns based on sorted SJEP-Tree." *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on*. IEEE, 2010.
- [15] Li, J., G. Dong and K. Ramamohanarao. 2001. Making use of the most expressive jumping emerging patterns for classification, *Knowledge and Information systems*, 3(2), pp. 131-145.
- [16] H. Fan and K. Ramamohanarao. 2002. An efficient single-scan algorithm for mining essential jumping emerging patterns for classification, *In Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, pp. 456-462.
- [17] P. Terlecki and K. Walczak. 2007. Jumping emerging patterns with negation in transaction databases-Classification and discovery, *Information Sciences*,177(24), pp. 5675-5690.
- [18] P. Terlecki and K. Walczak. 2008. Efficient discovery of top-k minimal jumping emerging patterns, *In Rough Sets and Current Trends in Computing*, Springer Berlin Heidelberg, pp. 438-447.
- [19] T. Gambin, and K. Walczak. 2009. Classification based on the highest impact jumping emerging patterns, *In Computer Science and Information Technology, IMCSIT'09, International Multiconference on*, pp. 37-42. IEEE.
- [20] Walczak. 2009. Classification based on the highest impact jumping emerging patterns, *In Computer Science and Information Technology, IMCSIT'09, International Multiconference on*, pp. 37-42. IEEE.
- [21] Kobylński, Ł. and K. Walczak. 2011. Efficient mining of jumping emerging patterns with occurrence counts for classification, *In Transactions on rough sets XIII*, Springer Berlin Heidelberg, pp. 73-88.
- [22] Liu, Q., P. Shi, Z. Hu, and Y. Zhang. 2014. A novel approach of mining strong jumping emerging patterns based on BSC-tree. *International Journal of Systems Science*, 45(3), pp. 598-615.
- [23] X. Shi, J. and H. Lei. 2008. A genetic algorithm-based approach for classification rule discovery. *International Conference on Information Management, Innovation Management and Industrial Engineering*, Vol. 1, pp. 175-178. IEEE..